1. What is encapsulation in C++?
   1. **A way to hide data and methods within a class**
   2. A way to expose all class members publicly
   3. A way to create multiple instances of a class
   4. A way to avoid using classes
2. Which access specifier allows class members to be accessed by derived classes?
   1. private
   2. **protected**
   3. public
   4. internal
3. Which C++ keyword is used for data hiding?
   1. **private**
   2. protected
   3. public
   4. encapsulate
4. What is encapsulation in C++?
   1. **A way to hide data and methods within a class**
   2. A way to expose all class members publicly
   3. A way to create multiple instances of a class
   4. A way to avoid using classes
5. Which access specifier allows class members to be accessed by derived classes?
   1. private
   2. **protected**
   3. public
   4. internal
6. What is polymorphism in C++?
   1. A way to create multiple instances of a class
   2. A way to hide data within a class
   3. **A way to provide multiple implementations for a function based on its arguments**
   4. A way to expose all class members publicly
7. Which type of polymorphism is achieved through function overloading?
   1. **Compile-time polymorphism**
   2. Runtime polymorphism
   3. Static polymorphism
   4. Dynamic polymorphism
8. Which keyword is used to implement method overriding in C++?
   1. extends
   2. **virtual**
   3. override
   4. over
9. Which type of polymorphism is achieved through function overriding?
   1. Compile-time polymorphism
   2. **Runtime polymorphism**
   3. Static polymorphism
   4. Dynamic polymorphism
10. What is abstraction in C++?
    1. A way to provide multiple implementations for a function based on its arguments
    2. A way to create multiple instances of a class
    3. **A way to hide the complexity of implementation and show only the necessary features**
    4. A way to expose all class members publicly
11. Which concept is closely related to abstraction in C++?
    1. Inheritance
    2. **Data hiding**
    3. Polymorphism
    4. Dynamic binding
12. Which keyword is used to declare an abstract class in C++?
    1. abstract
    2. base
    3. **virtual**
    4. pure\_abstract
13. What is inheritance in C++?
    1. A way to hide the complexity of implementation and show only the necessary features
    2. A way to create multiple instances of a class
    3. A way to achieve dynamic polymorphism
    4. **A way to create a new class by reusing properties of an existing class**
14. Which type of inheritance involves one base class and one derived class?
    1. **Single inheritance**
    2. Multiple inheritance
    3. Hierarchical inheritance
    4. Multilevel inheritance
15. Which keyword is used to derive a class publicly in C++?
    1. extends
    2. base
    3. inherits
    4. **public**
16. What is dynamic binding in C++?
    1. A way to create multiple instances of a class
    2. A way to achieve static polymorphism
    3. A way to determine the type of an object at runtime
    4. **A way to choose the implementation of a function at runtime**
17. Which type of polymorphism is achieved through dynamic binding?
    1. Compile-time polymorphism
    2. **Runtime polymorphism**
    3. Static polymorphism
    4. Dynamic polymorphism
18. Which keyword is used to call a virtual function through dynamic binding?
    1. dynamic\_call
    2. virtual\_call
    3. dynamic\_cast
    4. **None of the above**
19. What is polymorphism in C++?
    1. A way to create multiple instances of a class
    2. A way to hide data within a class
    3. **A way to provide multiple implementations for a function based on its arguments**
    4. A way to expose all class members publicly
20. Which type of polymorphism is achieved through function overloading?
    1. **Compile-time polymorphism**
    2. Runtime polymorphism
    3. Static polymorphism
    4. Dynamic polymorphism
21. Which keyword is used to implement method overriding in C++?
    1. extends
    2. **virtual**
    3. override
    4. over
22. What is abstraction in C++?
    1. A way to provide multiple implementations for a function based on its arguments
    2. A way to create multiple instances of a class
    3. **A way to hide the complexity of implementation and show only the necessary features**
    4. A way to expose all class members publicly
23. Which concept is closely related to abstraction in C++?
    1. Inheritance
    2. **Data hiding**
    3. Polymorphism
    4. Dynamic binding
24. What is inheritance in C++?
    1. A way to hide the complexity of implementation and show only the necessary features
    2. A way to create multiple instances of a class
    3. A way to achieve dynamic polymorphism
    4. **A way to create a new class by reusing properties of an existing class**
25. Which type of inheritance involves one base class and one derived class?
    1. **Single inheritance**
    2. Multiple inheritance
    3. Hierarchical inheritance
    4. Multilevel inheritance
26. Which keyword is used to derive a class publicly in C++?
    1. extends
    2. base
    3. inherits
    4. **public**
27. class A {

public:

virtual void display() { cout << "A"; }

};

class B : public A {

public:

void display() { cout << "B"; }

};

int main() {

A obj;

A \*ptr = new B;

obj.display();

ptr->display();

return 0;

}

1. **AB**
2. BA
3. AA
4. BB
5. class Base {

public:

void show() { cout << "Base"; }

};

class Derived : public Base {

public:

void show() { cout << "Derived"; }

};

int main() {

Derived d;

Base \*ptr = &d;

ptr->show();

return 0;

}

* 1. **Base**
  2. Derived
  3. Compile Error
  4. Undefined Behavior

1. class A {

public:

virtual void print() { cout << "A"; }

};

class B : public A {

public:

void print() { cout << "B"; }

};

int main() {

B obj;

A \*ptr = &obj;

ptr->print();

return 0;

}

1. A
2. **B**
3. Compile Error
4. Undefined Behavior
5. class Animal {

public:

void speak() { cout << "Animal"; }

};

class Dog : public Animal {

public:

void speak() { cout << "Dog"; }

};

int main() {

Dog d;

Animal \*ptr = &d;

ptr->speak();

return 0;

}

1. **Animal**
2. Dog
3. Compile Error
4. Undefined Behavior
5. class Shape {

public:

virtual void draw() { cout << "Shape"; }

};

class Circle : public Shape {

public:

void draw() { cout << "Circle"; }

};

int main() {

Shape \*shapes[2];

shapes[0] = new Shape;

shapes[1] = new Circle;

for (int i = 0; i < 2; ++i)

shapes[i]->draw();

return 0;

}

1. ShapeShape
2. CircleShape
3. **ShapeCircle**
4. CircleCircle
5. What is dynamic binding in C++?
   1. A way to create multiple instances of a class
   2. A way to achieve static polymorphism
   3. A way to determine the type of an object at runtime
   4. **A way to choose the implementation of a function at runtime**
6. Which type of polymorphism is achieved through dynamic binding?
   1. Compile-time polymorphism
   2. **Runtime polymorphism**
   3. Static polymorphism
   4. Dynamic polymorphism
7. Which keyword is used to call a virtual function through dynamic binding?
   1. dynamic\_call
   2. virtual\_call
   3. dynamic\_cast
   4. **None of the above**
8. class A {

public:

virtual void print() { cout << "A"; }

};

class B : public A {

public:

void print() { cout << "B"; }

};

int main() {

A \*a = new B;

B \*b = static\_cast<B \*>(a);

b->print();

return 0;

}

a) A

**b) B**

c) Compile Error

d) Undefined Behavior